Valley Seeking Clustering Based on Graph Theory

Aleksandra Krstić, Sanja Vujnović, and Željko Đurović

***Abstract*— Graph-theoretic algorithms for nonparametric clustering represent an important approach to data clustering. The paper considers one such noniterative algorithm which behaves similar to valley-seeking algorithm but requires less computational efforts. Same as some other nonparametric approaches, it is suitable for clustering of irregularly shape clusters in different metric spaces. It results in unimodal sets (trees) and its applicability is not limited to low dimension problems. Although it is considered easy to use, since it is governed by a single scalar which determines the considered neighborhood, its performance is highly dependent on the choice of this control parameter and the distance metric itself. This paper explores the dependency of the control scalar from data dimensionality and number of samples and considers the possibility of giving recommendations for the choice of this parameter for a preset metric which is suitable for a given dataset.**

***Index Terms*—Clustering, Nonparametric clustering, Valley seeking clustering, Graph-theoretic algorithms.**

I. Introduction

CLUSTERING analysis plays a critical role in various scientific and engineering domains, such as computer vision [1], bioinformatics [2], data mining [3], and many other machine learning disciplines. Its main goal is to group similar objects into clusters and there are two approaches to solving this task: parametric and nonparametric clustering.

Parametric clustering assumes that the data is generated from a specific statistical model with a fixed number of parameters, and the clustering algorithm attempts to estimate these parameters. In contrast, nonparametric clustering makes fewer assumptions about the data distribution and can handle more complex data structures. They are usually based on distance or density measures and do not assume a specific model for the data. This flexibility makes nonparametric clustering methods particularly useful for clustering high-dimensional or noisy data. However, nonparametric methods can be computationally more intensive and may require more tuning of parameters.

Valley seeking clustering is one of the oldest nonparametric clustering algorithms and it dates back to 1970s [4]. The main idea is simple and is based on identifying valleys in the sample density function of the data, and by moving away from the valleys it locates the cluster centers. Unlike many other clustering algorithms this approach does not need any prior information about the number of clusters and is able to identify clusters of arbitrary shapes.

Nonparametric valley seeking approach has been extensively studied in the literature in the past decades, and there are many variations of this method. Some studies have suggested using it in combination with other clustering techniques, such as k-means to further refine its accuracy, while other studies tend to incorporate some prior knowledge about the data structure [5]. Graph-theoretic approach to valley seeking [6] is introduced as a way to reduce computational complexity of the algorithm. It is a noniterative procedure based on graph theory which results in unimodal sets or trees that represent specific clusters.

One major issue with valley seeking approach to clustering is that it is heavily dependent on the choice of control parameter that dictates how big is the analyzed area around each sample. This paper explores how this control scalar affects the clustering results in several ways. First, the performance of the algorithm is tested for simple set of clusters when the number of samples in clusters changes. After that the algorithm is tested on three different scenarios when the control parameter changes. Finally, the possibility of adaptive approach for the choice of this parameter is analyzed and discussed.

This paper is structured as follows. In Section II the theoretical background to the algorithm is described. Section III illustrates the experimental setup, while the results are shown in Section IV. The conclusion to the paper is written in Section V.

II. Theoretical Analysis

A. Valley Seeking

The valley seeking approach to clustering assumes that probability density function of the samples can be parametrized with peaks and valleys. Peaks represent centers of clusters, while valleys are the boundaries between them. The peaks are detected by observing the gradient of the probability density function and by moving the sample toward the direction of the gradient, thus ‘climbing’ away from the valley and towards the peak. Hence, two issues need to be addressed in order to successfully implement this procedure: how to estimate the gradient of the probability density function and how to use this information to form clusters.

Aleksandra Krstić is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: akrstic@etf.bg.ac.rs)

Sanja Vujnović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: svujnovic@etf.bg.ac.rs)

Željko Đurović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: zdjurovic@etf.bg.ac.rs).

 ISBN 978-86-7466-970-9

There are many suggestions in the literature on how to estimate the gradient of the density function and following is the procedure suggested by Fukunaga [7]. In order to calculate the gradient at $X \in R^n$ we will form a region $\Gamma(X)$ which has a radius of r:

$$\Gamma(X) = \{Y: d(Y,X) \leq r\}, \quad (1)$$

where $d(Y,X)$ is a distance measure

$$d^2(Y,X) = (Y-X)^T A^{-1}(Y-X), \quad (2)$$

and the matrix A is a metric used to measure the distance.

Having defined this, we can introduce the expected vector Y from the region $\Gamma(X)$ which is called the local mean, M_L, and is calculated as

$$\mathrm{M_L}(X) = E\{(Y-X)|\Gamma(X)\} = \int_{\Gamma(\mathrm{X})} (Y-X)\frac{f(Y)}{u_0}dY. \quad (3)$$

Here $f(\cdot)$ is probability density function of the samples and u_0 is the probability that the sample Y is in the region $\Gamma(X)$

$$u_0 = \int_{\Gamma(X)} f(Y)dY \approx f(X)v, \quad (4)$$

where v is the volume of $\Gamma(X)$. Now the function $\Gamma(X)$ can be expanded around X in a Taylor series obtaining:

$$f(Y) \approx f(X) + (Y-X)^T \nabla f(X). \quad (5)$$

Substituting eq. (4) and (5) into (3), as done in [7] an expression for normalized gradient is obtained:

$$\frac{\nabla f(X)}{f(X)} \approx \frac{n+2}{r^2} A^{-1} M_L(X). \quad (6)$$

The eq. (6) demonstrates how it is possible to obtain the gradient normalized by the probability density function. That has significant advantages to estimating the gradient itself, due to the fact that when the gradient is estimated in the valleys in which it has smaller value, the signal is amplified by dividing it with the probability density function which kas smaller values as well, and the normalized value is easier to estimate.

After obtaining the normalized gradient, clustering is conducted by moving the samples in the direction $\rho M_L(X)$, where ρ is the step parameter. The idea is quite intuitive, but the implementation of the entire process is computationally expensive because after each step the entire data set is changed. For this reason, the non-iterative procedure based on graph theory is suggested.

B. Graph Theoretic Approach

Graph theoretic approach to valley seeking avoids iterative operation by forming trees in which each sample is a node. Each node initiates a branch pointing at another node which is called a predecessor. A series of these branches is called a directed path, keeping in mind that there is no direct path from a node to itself. At the top of the tree there is a final node which does not have a predecessor, and it is called the root of the tree. Every node, except the final node, has exactly one predecessor, and each node can be a predecessor of zero, one or multiple nodes. This structure is called a directed tree and the main idea is to determine a predecessor to each node as a node which is in the direction of the steepest gradient.

It has been shown [7] that each node X_j acquires its predecessor X_k from the region $\Gamma(X_j)$ so that the following equation is satisfied

$$s_{kj} = \max_{X_l \in \Gamma(X_j)} s_{lj}. \quad (7)$$

This means that the predecessor of X_j is a node from the region $\Gamma(X_j)$ which has the steepest gradient to the node X_j. Here the steepness between nodes X_j and X_l is calculated as follows:

$$s_{lj} = \|\nabla f(X_j)\| \cos(\theta_{lj}), \quad (8)$$

where θ_{lj} is the angle between vectors $\nabla f(X_j)$ and $(X_l - X_j)$. Seeing how vectors $\nabla f(X_j)$ and $M_L(X_j)$ have the same direction, and how $\|\nabla f(X_j)\|$ is the same for all the nodes in the region $\Gamma(X_j)$, it is easy to conclude that the maximal steepness can be calculated as a minimal angle between vector $M_L(X_j)$ and vector $(X_l - X_j)$.

Following the logic described previously, for each node X_j the maximal steepness s_{kj} is calculated, and one of the following rules is applied:

1) If $s_{kj} > 0$: X_k is a predecessor of X_j;

2) If $s_{kj} < 0$: X_j is a root of the tree;

3) If $s_{kj} = 0$: form the set $\Pi(X_j) = \{X_k | X_k \in \Gamma(X_j), s_{kj} = 0\}$ and eliminate from the set all the nodes X_k from which directed paths toward X_j already exist. If resulting set $\Pi(X_j)$ is empty, then X_j is a root of the tree. Otherwise, predecessor to X_j is X_t such that

$$\|X_t - X_j\| = \min_{X_k \in \Pi(X_j)} \|X_k - X_j\|. \quad (9)$$

This way all the observations which have the common root of the tree correspond to the same cluster, and there are as many clusters as there are roots.

III. Experimental Setup

Valley seeking algorithm with its implementation based on graph theory, as described in Section II, has many advantages. Most notably fast execution time (due to non-iterative nature of the implementation) and ability to detects clusters of irregular shapes in a high dimensional space (due to the nonparametric nature of valley seeking and the fact that no prior knowledge of the distribution is assumed). The greatest drawback, however,

 ISBN 978-86-7466-970-9

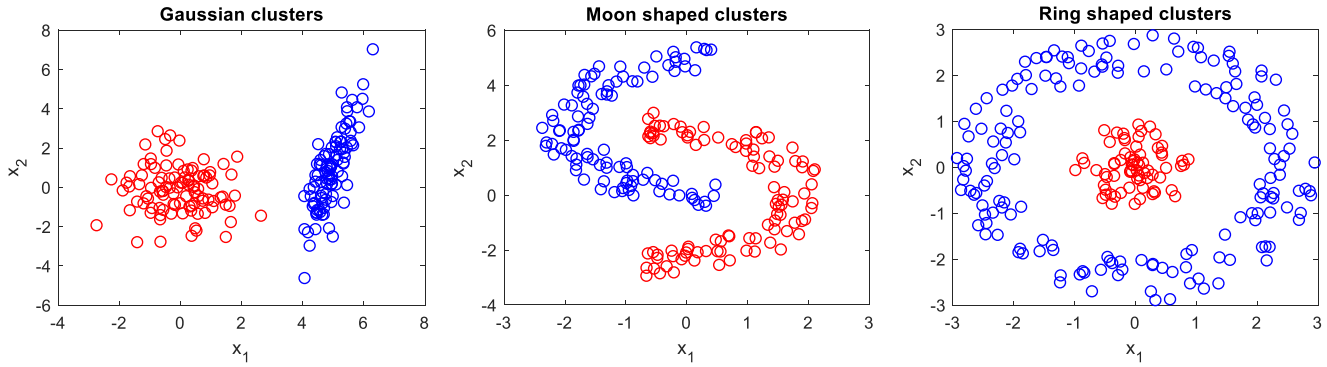

Fig. 1. First (left), second (middle) and third (right) clustering scenarios. First cluster is shown in blue color, while second cluster is shown in red.

is the fact that the performance of this algorithm is heavily dependent on the control variable r from eq. (1) which dictates the area of the region which is considered when estimating the direction of the steepest ascent.

The main idea is to explore the dependency of the control variable r with respect to the number of samples and the shapes of the cluster and to consider recommendations for the choice of this parameter. Three different scenarios will be considered and, for easier visual representation, all three will be in 2-dimensional space. The first scenario consists of two Gaussian clusters linearly separable (Fig. 1, left), the second consists of two nonlinearly separable moon shaped clusters (Fig. 1, middle) and the third scenario has one cluster completely immersed within the second ring shaped cluster (Fig. 1, right).

Three experiments will be conducted, all of which will, for simplicity, adopt A form eq. (2) to be a unit matrix:

Experiment 1: For the first set of clusters use fixed control parameter, r, and change the number of samples within each cluster.

Experiment 2: For all three sets of clusters from Fig. 1 change the value of the control parameter and analyze how the accuracy changes.

Experiment 3: Suggest alternative for calculating control parameter and analyze the accuracy of clustering.

IV. Results

A. Experiment 1

This is an illustrative experiment which is used to test to what extent does the number of observations influence the accuracy of the algorithm for a fixed control parameter r. This has been tested on the first set of clusters (Fig. 1, left) with the value of $r = 1$ and $r = 2$, for the number of samples in each cluster $N \in \{10,20,40,80,150\}$. It is logical to assume that the greater the density of observations (ie. the more samples there are in each cluster) the bigger the accuracy. There is concern that too many samples in a given window can average out the peaks and the valleys so, theoretically, there should be the upper limit after which the increase of N does not increase accuracy.

The experiment is conducted such that for each pair of values r and N, 100 scenarios are generated with different samples, and an average of those is observed. The average number of clusters is given in Fig. 2 (up), while the percentage of time the algorithm has correctly detected that there are 2 clusters is given in Fig. 2 (down). Since the clusters are linearly separable, in all the cases in which the algorithm has detected 2 clusters, the classification is 100% accurate (i.e. each observation corresponds to the correct cluster).

It is evident that for $r = 1$ the accuracy of the algorithm is poor regardless of the number of observations which are generated. It somewhat improves as the parameter N increases, but even for the highest value of N tested, the algorithm converges to 2 clusters only 5% of the time. For $r = 2$, on the other hand, the behavior of the algorithm is as expected. The accuracy increases as N increases and, in these simulations at least, the upper bound to N has not been determined.

The results of this experiment corroborate the premise of the problem stated in this paper – the accuracy of the valley seeking clustering algorithm is highly dependent on the value of r, and for the values of r that are poorly chosen, no amount of increase of samples can improve the algorithm.

B. Experiment 2

In this experiment all three clustering scenarios from Fig. 1 were tested for a fixed number of observations, N, and variable control parameter, r. The number of samples in each cluster for the first and second scenario is the same, $N = 100$, while in the third scenario, the cluster which corresponds to the larger area (blue one in Fig. 1) has 170 samples, while the one that corresponds to the smaller area has 70 samples. The experiment is tested for values $r \in \{1,1.25,1.5,1.75,2\}$ and an average result of 60 simulation per parameter per scenario is shown in Fig 3.

As can be inferred from the first experiment, in the first scenario with linearly separable clusters, the higher the value of r, the better the clustering algorithm behaves. There is probably an upper limit after which this increase in performance stops, but it is not detected in the tested set of values. As for the second and the third scenario, they both tend to make large errors for $r \in \{1,1.25\}$ by detecting, on average, more than 2 clusters, and they tend to make large errors for $r \in \{1.75,2\}$ by detecting, on average, only one cluster. Here we can see that there is indeed an optimal value for the control parameter, and it is $r = 1.5$. Also, the ring shaped scenario seems to be more challenging for valley seeking algorithm, as the accuracy tends to be higher for the moon shaped one for all values of r.

 ISBN 978-86-7466-970-9

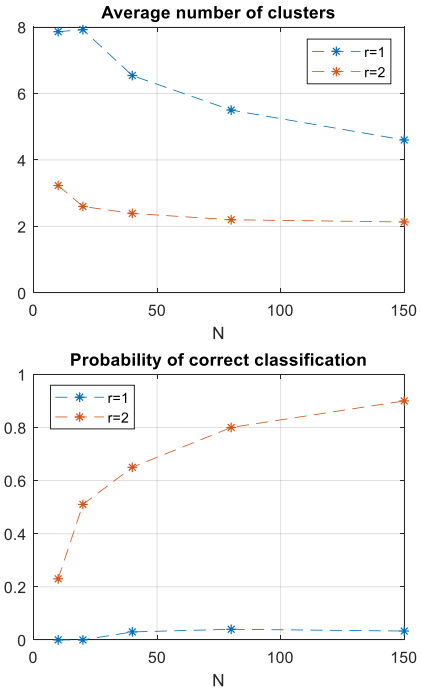

Fig. 2. Average number of clusters (up) and probability of correct classification (down) for different number of samples, based on 100 simulations for the first clustering scenario.

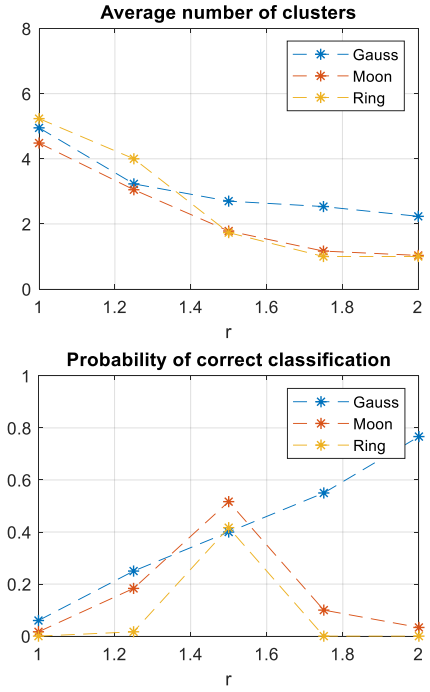

Fig. 3. Average number of clusters (up) and probability of correct classification (down) for different values of control coefficient, based on 60 simulations for each clustering scenario.

As can be inferred from the first experiment, in the first scenario with linearly separable clusters, the higher the value of r, the better the clustering algorithm behaves. There is probably an upper limit after which this increase in performance stops, but it is not detected in the tested set of values. As for the second and the third scenario, they both tend to make large errors for $r \in \{1,1.25\}$ by detecting, on average, more than 2 clusters, and they tend to make large errors for $r \in \{1.75,2\}$ by detecting, on average, only one cluster. Here we can see that there is indeed an optimal value for the control parameter, and it is $r = 1.5$. Also, the ring shaped scenario seems to be more challenging for valley seeking algorithm, as the accuracy tends to be higher for the moon shaped one for all values of r.

C. *Experiment 3*

In this experiment we entertain the possibility of implementing adaptive control parameter r which will grow larger as the number of samples in the region becomes sparse and grows smaller as the density of samples increases. One way to approach this issue is to aim to adjust the control parameter for each node, so that the number of samples in the region $\Gamma(X_j)$ remains constant. In this way, instead of predetermining the value of r, we need to adopt the value n which represents the number of observations within $\Gamma(X_j)$. It is intuitively clear that the parameter n should be proportional to the standard deviation of clusters and that it should in some way reflect the local statistics of the samples as well. Seeing how we already have a prior knowledge of the performance of valley seeking algorithm for different values of r from the previous experiment, it can be concluded that for $r = 1.5$ there is the best performance in second and third clustering scenario. For that value of r, on average, there are between 30 and 40 samples in each region $\Gamma(X_j)$.

Table I shows the results of adaptive step implementation when the number of samples in each gate is $n = 35$ and comparison with the best results achieved in the previous experiment are given as well. The accuracy corresponds to the number of times clustering algorithm has correctly detected 2 clusters, and as we can see, the adaptive choice of r yields better results for all scenarios.

The results seem promising; however, the choice of one parameter, r, is now substituted with the choice of another, n. Figure 4 shows how the performance of adaptive control parameter approach changes with n and it is clear that even though the results are better than in experiment 2, the accuracy still deeply depends on the value of this parameter.

TABLE I
COMPARISON OF THE BEST RESULT FROM EXPERIMENT 2 WITH THE RESULTS OF EXPERIMENT 3 WITH PARAMETER $n = 35$.

Best results for experiment 2 – fixed r			
	Gauss	Moon	Ring
r	2	1.5	1.5
Accuracy	0.8	0.5	0.4
Results for experiment 3 – adaptive r			
	Gaus	Moon	Ring
r_{min}	0.76	0.97	0.46
r_{max}	4.48	2.68	2.15
Accuracy	1	0.9	0.6

 ISBN 978-86-7466-970-9

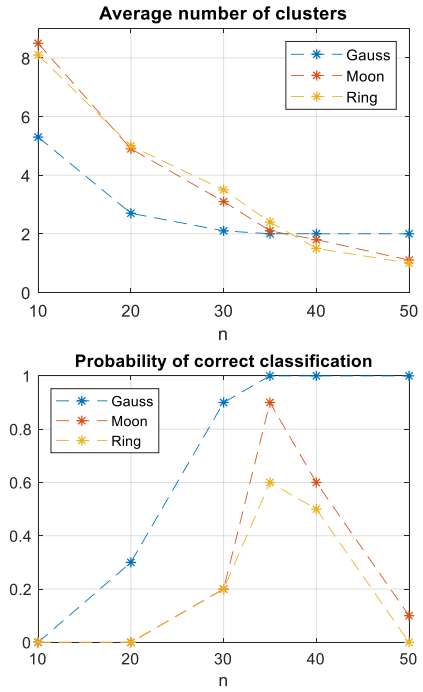

Fig. 4. Average number of clusters (up) and probability of correct classification (down) for different values of parameter n, based on 10 simulations for each clustering scenario.

V. Conclusion

Valley seeking clustering is a well-established nonparametric method that can identify clusters of arbitrary shapes without prior knowledge about the number of clusters. However, the performance of the algorithm is highly dependent on the choice of control parameter which determines the size of analyzed area around each sample.

This paper has explored how this control scalar affects the clustering results in various scenarios and proposed an adaptive approach for selecting this parameter. The results show that the proposed approach can improve the accuracy of valley seeking clustering and make it more robust to different data characteristics.

In future work, the adaptive approach should be further refined, by stating clear rules for the change of control parameter or the choice of new parameter n. Furthermore, the application to other clustering algorithms should be explored. Overall, this study contributes to the understanding and improvement of nonparametric clustering methods, which can have important applications in data mining, machine learning, and pattern recognition.

Acknowledgment

This work was financially supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia under contract number: 451-03-47/2023-01/200103.

References

[1] X. Ji, J. F. Henriques, A. Vedaldi, “Invariant information clustering for unsupervised image classification and segmentation,” Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9865-9874, 2019.

[2] R. Petegrosso, Z. Li, R. Kuang, “Machine learning and statistical methods for clustering single-cell RNA-sequencing data,” *Briefings in bioinformatics*, vol. 21, no. 4, pp. 1209-1223, 2020.

[3] Y. Sato, K. Izui, T. Yamada, S. Nishiwaki, “Data mining based on clustering and association rule analysis for knowledge discovery in multiobjective topology optimization,” *Expert Systems with Applications*, vol. 119, pp. 247-261, 2019.

[4] W. L. Koontz, K. Fukunaga, “A nonparametric valley-seeking technique for cluster analysis,” *IEEE transactions on computers*, vol. 100, no. 2, pp. 171-178, 1972.

[5] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, A. A. Akinyelu, “A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects,” *Engineering Applications of Artificial Intelligence* vol. 110, pp.104743, 2022.

[6] W. L. G. Koontz, P. M. Narendra, K. Fukunaga, “A graph-theoretic approach to nonparametric cluster analysis,” *IEEE Transactions on Computers*, vol. 25, no. 09, pp. 936-944, 1976.

[7] K. Fukunaga, *Introduction to statistical pattern recognition*, 2nd ed. Elsevier, 2013.

 ISBN 978-86-7466-970-9